01 May 2006

Time-Delay Neural Network Based Predictions of Elephant Distribution in a South African Game Reserve

Parviz Palangpour

Ganesh K. Venayagamoorthy
Missouri University of Science and Technology

Kevin Duffy

Time-Delay Neural Network Based Predictions of Elephant Distribution in a South African Game Reserve

Parviz Palangpour[1], *Student Member, IEEE,* Ganesh K. Venayagamoorthy[1], *Senior Member, IEEE* and Kevin Duffy[2]
[1]Real-Time Power and Intelligent Systems Laboratory.
Dept. of Electrical and Computer Engineering
University of Missouri - Rolla
Rolla, MO 65409, USA
pmpv3b@umr.edu, gkumar@ieee.org
[2]Centre for Systems Research
Durban Institute of Technology, Durban, South Africa

***Abstract*— A large portion of South Africa's elephant population can be found on small wildlife reserves. When confined to enclosed reserves the elephant densities are much higher than observed in the wild. The large nutritional demands and destructive foraging behavior of elephants threaten rare species of vegetation. If conservation management is to protect threatened species of vegetation, knowing how long elephants will stay in one area of the reserve as well as which area they will move to next is essential. The goal of this study is to train an artificial neural network to predict an elephant herd's next position in the Pongola Game Reserve. Accurate predictions would provide a useful tool in assessing future impact of elephant populations on different areas of the reserve. The particle swarm optimization (PSO) algorithm is used to adapt the weights of the neural network. Results are presented to show the effectiveness of TDNN-PSO for elephant distribution prediction.**

I. Introduction

Converting farmland into small wildlife reserves is becoming common throughout South Africa. These reserves introduce the major species that attract tourists: lion, leopard, elephant, rhino, giraffe and hippopotamus. The last four are often classified as *mega* herbivores due to the large size and great nutritional demand of the species. It is known that several mega herbivores, such as the elephant, are destructive foragers and their overpopulation can adversely impact the ecology of their range. Elephants are also known to exhibit preferences for certain species of vegetation [1]. As a result, elephants confined to small areas can have a heavy impact on the diversity of vegetation throughout their habitat.

The neural network (NN) has been applied successfully to many problems involving time series prediction and modeling of non-linear systems [2]-[3]. The time-delay neural network (TDNN) is a feedforward neural network capable of using a fixed number of previous system inputs to predict the following output of the system. The TDNN has been used extensively for speech recognition and has been shown to perform quite well [4]-[7]. In this paper, the TDNN is used to predict short term elephant herd movement using only previous positions. The particle swarm optimization (PSO) is applied for training two TDNNs, one network predicts the x coordinate and the other predicts the y coordinate of the herds position.

This paper is organized as follows. Section II describes the game reserve from which data was collected. Section III explains the TDNN architecture. Section IV presents the techniques used to train the TDNN to predict elephant distributions. Finally, Section V analyzes the results obtained from the study and offers some future research directions.

II. Case Study

Located on the southeastern border of Swaziland in South Africa is the Pongola Game Reserve. In June 1997, a family group of 17 elephants from another park were introduced to the reserve. While the reserve is 73.6 km^2, the same family group had grown to 37 individuals by January 2004. Three bulls which move independently of each other also live on the reserve. In addition to the family group and bulls, a group of five young elephants also shared the habitat during this period. Each of these three groups migrate to different areas of the reserve as separate herds. A study has indicated at least one rare tree, the *Sclerocarya birrea* is being removed at a rate higher than annual regeneration [1]. The diversity of vegetation in the reserve can be seen in Fig. 1.

A. Data Collection

In February 2000, a cow from the family group and a bull were fitted with a GPS satellite collar to monitor their movement. The GPS positions of both elephants were recorded on semi-regular intervals until March 2002. Because elephants in a family herd move as a group, the movement of a single elephant represents the general movement of the herd.

B. Pretreatment of Data

The longitude and latitude coordinates from each set of recorded positions are first transformed to a projected cartesian coordinate system. The data from each group is

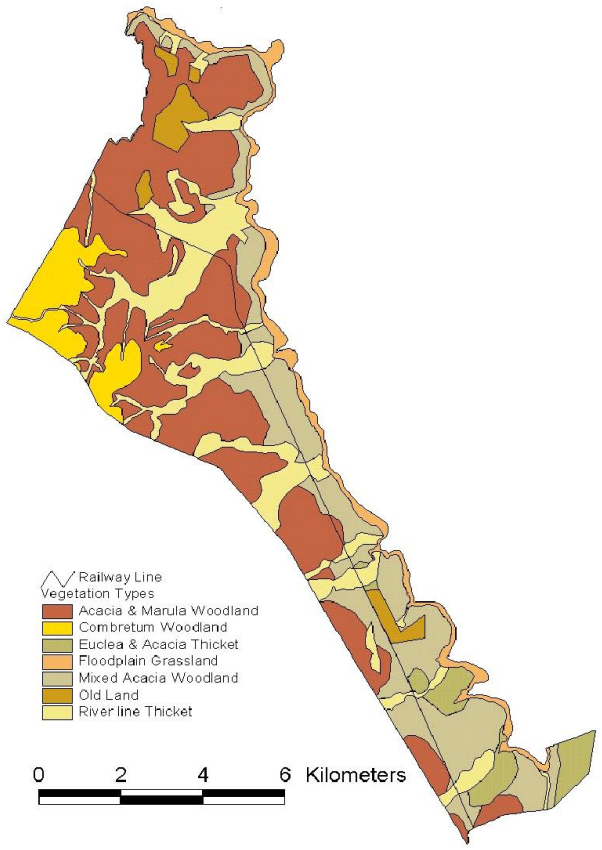

Fig. 1. The Pongola Game Reserve

then interpolated to produce a dataset consisting of one position for every twenty-four hour interval. All positions are normalized to be in [-1,1]. Two separate sets of data are extracted to be used for training and validation, the first set uses the first year of data for training and the second year of data for validation. Based on a previous study of elephant behavior that suggested elephant movement is seasonal, a different approach is used for the second version of the training and validation set; each dataset is split into two seasons, summer and winter, for each of the two years [1]. The first season of the first year is used to train the network while the data for same season during the second year is used as a testing dataset.

III. TDNN for Elephant Distribution Prediction

The TDNN, a variant of the multilayer perceptron (MLP), uses time-delayed inputs to the hidden layers. In the hidden layer each neuron is presented a total number of τ delayed values in addition to the current value, for each input to the network. In this experiment, the network had only one hidden layer, so each neuron in the hidden layer would be presented x_i(t), x_i(t-1),..., x_i(t-τ) for each input i.

For example, let each input x_i(t) to the network become a τ + 1 dimensional vector $\mathbf{x}_i$(t)

$$\mathbf{x}_i(t) = [x_i(t), x_i(t-1), ..., x_i(t-\tau)] \quad (1)$$

Each weight w_{ji} then becomes a τ + 1 dimensional vector $\mathbf{w}_{ji}$

$$\mathbf{w}_{ji} = [w_{ji}(0), w_{ji}(1), ..., w_{ji}(\tau)] \quad (2)$$

where $\mathbf{w}_{ji}$ is the weight vector for the connection between the input i and the neuron j.

All neurons in the hidden layer used the hyperbolic tangent activation function (3). The output d_j(t) of each neuron j is

$$d_j(t) = \frac{(1 - e^{-a_j(t)})}{(1 + e^{-a_j(t)})} \quad (3)$$

where

$$a_j(t) = \sum_{t=1}^{\tau+1} s_{ji}(t) \quad (4)$$

and

$$s_{ji}(t) = w_{ji}^T x_i(t) \quad (5)$$

The output neuron uses a linear activation function so the output of the network $z(t)$ is

$$z(t) = \sum_{n=1}^{N+1} v_n d_n(t) \quad (6)$$

which is the network's prediction of $x(t+1)$. A TDNN using two hidden neurons with $\tau = 2$ is shown in Fig. 2. The network is presented the values of $x(t)$, $x(t-1)$ and $x(t-2)$ then it predicts the value of $x(t+1)$ where $x(t)$ is the x coordinate of the herd position at time t.

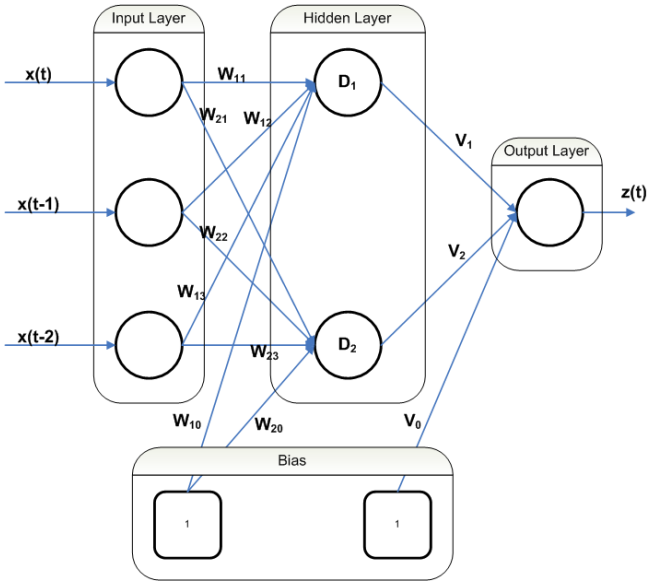

Fig. 2. A TDNN with τ=2 and two hidden neurons.

IV. Training The TDNN

The TDNN is trained by presenting a number of patterns from one portion of the dataset and adjusting the weights until the error is acceptable or becomes stagnant; this simply means training stops when the accuracy of the TDNN predictions are better than a predetermined error interpreted to be 'good enough', or the accuracy does not improve with more training. There are a number of different training algorithms used for this purpose, though backpropagation and other forms of gradient descent have been used the most extensively. In addition, evolutionary algorithms such as PSO and genetic algorithms have also been used to optimize the weights of NN's [8]-[9]. When used to train the same NN, comparisons of backpropagation, genetic algorithms and PSO have shown that PSO requires the fewest number of training iterations to achieve the same error [10]-[11]. Particle swarm optimization is selected because it is computationally efficient and has been shown to perform well on a large variety of problems [12].

Two separate networks are needed to predict the position (x,y), the network that predicts the x position (NN_x) and the network that predicts the y position (NN_y). Attempts are made to use both the x and y coordinates to train each network, but this did not prove feasible.

A. Particle Swarm Optimization

Particle swarm optimization is used to find the optimal weights for the TDNN. The PSO algorithm was developed by Kennedy and Eberhart and is based on the evolution of a population of particles [13]. Each particle in the population has a position vector which represents a potential solution to the problem. The particles are initialized to random positions throughout the search space and for each iteration of the algorithm a velocity vector is computed and used to update each particles position. Each particles velocity is influenced by the particles own experience as well as the experience of its neighbors.

In this study, the *global* version of the PSO algorithm is applied. At each time step t, a cost function $f(t)$ is used to measure the fitness of each particle i in the population. The best position each particle attained is stored in the vector p_i, while the best position attained by any particle in the population is stored in the vector p_g. The velocity vector $v_i(t)$ for each particle is then updated.

$$\begin{aligned} v_i(t+1) = v_i(t) &+ c_1\rho_1(p_i - x_i(t)) \\ &+ c_2\rho_2(p_g - x_i(t)) \end{aligned} \tag{7}$$

where c_1 and c_2 are positive and ρ_1 and ρ_2 are uniformly distributed random numbers in [0,1]. The term c_1 is called the cognitive term and c_2 is called the social term. These two values balance the influence between the particles own best performance and that of the population. The velocity is constrained between the parameters V_{min} and V_{max} to limit the maximum change in position.

$$v_i(t+1) = \begin{cases} v_{Max} & \text{if } v_i\text{(t + 1)} > V_{max} \\ v_{Min} & \text{if } v_i\text{(t + 1)} < V_{min} \\ v_i(t+1) & \text{else} \end{cases} \tag{8}$$

The position of each particle is then updated using the new velocities.

$$x_i(t+1) = x_i(t) + v_i(t+1) \tag{9}$$

The position in each dimension is limited between the parameters X_{min} and X_{max}.

$$x_i(t+1) = \begin{cases} x_{Max} & \text{if } x_i\text{(t + 1)} > X_{max} \\ x_{Min} & \text{if } x_i\text{(t + 1)} < X_{min} \\ x_i(t+1) & \text{else} \end{cases} \tag{10}$$

B. Constriction Coefficient

Clerc and Kennedy later suggested the use of a 'constriction coefficient' to improve the convergence characteristics of PSO [14]. This version of the constriction model is referred to as *Type 1* Constriction and only the velocity equation is changed. The new velocity equation is:

$$\begin{aligned} v_i(t+1) = \chi[v_i(t) &+ c_1\rho_1(p_i - x_i(t)) \\ &+ c_2\rho_2(p_g - x_i(t))] \end{aligned} \tag{11}$$

The value for χ is computed as

$$\chi = \frac{2\kappa}{|2 - \alpha - \sqrt{\alpha(\alpha-4)}|} \tag{12}$$

where

$$4 < c_1 + c_2 \tag{13}$$

$$\alpha = c_1 + c_2 \tag{14}$$

$$\kappa \in [0, 1] \tag{15}$$

This velocity equation is very similar to (7), although the term $v_i(t)$ no longer has a coefficient of 1, but now χ. The parameter κ influences the speed of convergence, using a larger value results in a slower convergence which allows the particles to explore the search space more throughly. For this experiment, κ is set to 1, which offers the slowest convergence.

C. Fitness Function

For the purpose of assessing the elephant impact on clearly defined areas of vegetation, the network with the least mean square error (MSE) of predictions would be more desirable than the network with the least average error. The fitness of particle p_i is

$$E = f(x_i) = \frac{1}{|P|}\sum_{t=1}^{|P|}(z(t) - P(t))^2 \tag{16}$$

where P is the dataset of training patterns, $z(t)$ is the output of the NN and $P(t)$ is the target output. Using the MSE as the fitness function will put emphasis on removing errors in the predictions that are greater than the mean error.

D. Training Parameters

In this paper, the cognitive and social terms, c_1 and c_2, are both set to 2.05; thus the cognitive term and social term have an equal probability of influencing the particle more. This results in a α of 4.1 and the value for χ is then 0.7298. The values of the weights and velocities are constrained to be in [-1,1]. The algorithm trained the networks for 500 iterations. The number of delayed inputs and neurons in the hidden layer were found empirically.

V. Results

The optimal value for τ in (1) is found to be 2 for both NN_x and NN_y. While NN_x achieves the lowest MSE using 3 hidden neurons, NN_y achieves the lowest MSE using 19 hidden neurons. This large difference in the number of hidden neurons is due to the difference in length of the x axis and y axis of the game reserve as can be seen in Fig. 1.

Using the first year of recorded data to train the networks and the second year of recorded data for testing offered much better results than the networks trained and tested against seasonal data. Tables I and II compare the MSE of each network using the two different pairs of training and testing datasets. It seems the NN is able to achieve better generalization using an entire year for training even if the data consists of the elephants migrating in different seasons. Figs. 3 and 4 compare the predictions of the two neural networks against the training dataset using the first and second year of data for training and testing respectively. The predictions of the same networks against the testing dataset are shown in Figs. 5 and 6.

TABLE I

Results using year 1 to train the networks and year 2 for testing

NN	Training MSE (km^2)	Testing MSE (km^2)
NN_x	0.3997	0.8133
NN_y	1.8820	3.6814

TABLE II

Results using the first season of the first year for training and the first season of the second year for testing

NN	Training MSE (km^2)	Testing MSE (km^2)
NN_x	0.4790	0.8985
NN_y	2.1378	5.2453

The results indicate that short term prediction is feasible. While the networks did not achieve great accuracy, much can be done in terms of network architecture, training and the amount of data used for training. In this experiment, the networks are trained using only the past positions, though elephant are likely to base movement on a large number of environmental variables. As vegetation is exhausted in different areas of the reserve, the elephants will no longer behave as they did when the vegetation was still available. This makes it very difficult to make accurate predictions without knowing the current availability of vegetation throughout the reserve.

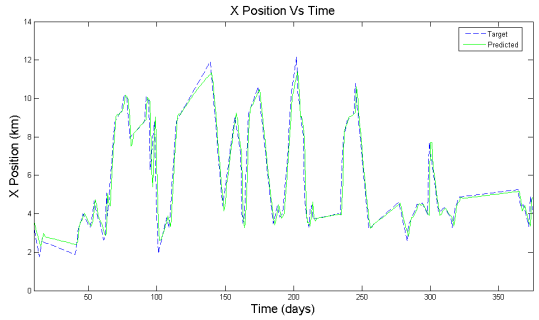

Fig. 3. Comparison of NN_x training results against the recorded data

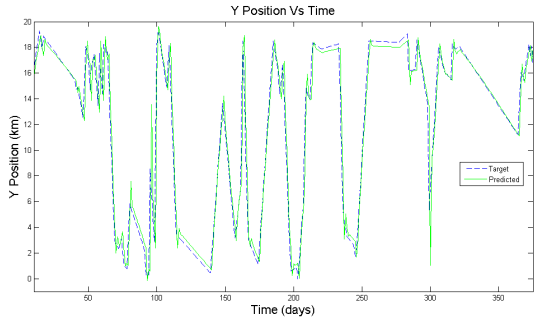

Fig. 4. Comparison of NN_y training results against the recorded data

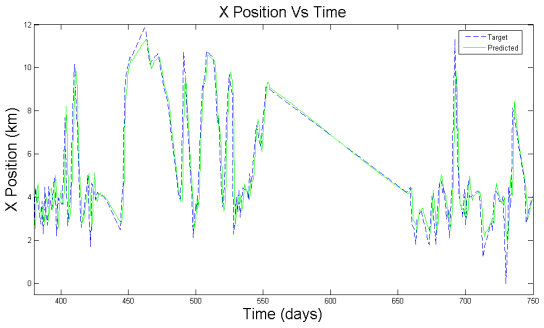

Fig. 5. Comparison of NN_x testing results against the recorded data

VI. Conclusions

A short term prediction system for elephant movement in a South African game reserve has been presented. TDNN trained with PSO have been shown to provide some degree of success, though research in neural networks and optimization algorithms are constantly progressing. Better results are surely attainable given the inherent flexibility of neural networks.

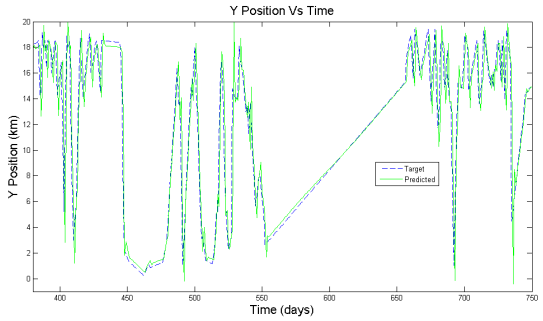

Fig. 6. Comparison of NN_y testing results against the recorded data

This work is only meant as a starting point, there are still many more tests and much work that needs to be done to fully examine the potential of this prediction system. Networks with feedback, such as recurrent neural networks, are capable of modeling much more complex dynamic systems and might be better suited to elephant distribution prediction.

REFERENCES

[1] G. Shannon, B. Page, K. Duffy, and R. Slotow "African elephant home range and habitat selection in Pongola Game Reserve, South Africa," Submitted to *African Journal of Ecology*.

[2] D. Shi, H. Zhang, and L. Yang, "Time-delay neural network for the prediction of carbonation tower's temperature," *IEEE Trans. Instrumentation and Measurement*, vol.52, pp.1125-1128, August. 2003.

[3] E.W. Saad, D.V. Prokhorov, and D.C. Wunsch, II, "Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks," *IEEE Trans. Neural Networks*, vol.9, pp.1456-1470, November 1998.

[4] H. Sawai, "TDNN-LR continuous speech recognition system using adaptive incremental TDNN training," International Conference on Acoustics, Speech, and Signal Processing, vol.1, pp.53-56, April 14-17, 1991.

[5] G.L. Berger and J.N. Gowdy,"TDNN based speaker identification," Proceedings of the Southeastern Symposium on System Theory, pp.396-399, March 7-9, 1993.

[6] C. Dugast, L. Devillers, and X. Aubert, "Combining TDNN and HMM in a hybrid system for improved continuous-speech recognition," *IEEE Trans. Speech and Audio Processing*, vol.2, pp.217-223, January 1994.

[7] M. Sugiyama, H. Sawai, and A.H. Waibel, "Review of TDNN (time delay neural network) architectures for speech recognition," IEEE International Symposium on Circuits and Systems, vol.1 pp.582-585, June 1991.

[8] D. Srinivasan, W.H. Loo, and R.L. Cheu, "Traffic Incident Detection Using Particle Swarm Optimization," Proceedings of the 2003 IEEE Swarm Intelligence Symposium, pp.144-151, April 24-26, 2003.

[9] S.H. Ling, F.H.F. Lenung, H.K. Lam, and P.K.S. Tam, "Short-Term Daily Forecasting in an Intelligent Home with GA-Based Neural Network," Proceedings of the 2002 International Joint Conference on Neural Networks, pp.997-1001, May 12-17, 2002.

[10] E.A. Grimaldi, F. Grimaccia, M. Mussetta, and R.E. Zich, "PSO as an effective learning algorithm for neural networks applications," Proceedings of the 2004 3rd International Conference on Computational Electromagnetics and Its Applications, pp.557-560, Nov 1-4, 2004.

[11] V. Gudise and G. Venayagamoorthy, "Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks," Proceedings of the 2003 IEEE Swarm Intelligence Symposium, pp.110-117, April 24-26, 2003.

[12] R. Eberhart; Y. Shi, "Particle swarm optimization: developments, applications and resources," Proceedings of the Congress on Evolutionary Computation, vol.1, pp.81-86, 2001.

[13] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of the International Conference on Neural Networks, vol.4, pp.1942-1948, Nov/Dec 1995.

[14] M. Clerc and J. Kennedy, "The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space," *IEEE Trans. Evolutionary Computation*, vol.6 pp.58-73, February 2002.